Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) A processing system comprising:

a plurality of first interrupts generated by a core, said plurality of first interrupts having programmable priorities;

a plurality of second interrupts that are generated external to said core, said second interrupts having architecturally fixed interrupt priorities;

a status register, coupled to said core, said status register comprising ~~having~~ a vector table, and an interrupt register, said interrupt register coupled to said vector table, said interrupt register having a plurality of configurable priority registers for storing said programmable priorities~~, said programmable priorities being different than those architected for said plurality of second interrupts~~; and

a priority encoder, coupled to both said first interrupts and to said second interrupts, and to said status register, said priority encoder prioritizing said first and second pluralities of interrupts utilizing said programmable priorities for said first interrupts and said architecturally fixed interrupt priorities for said second interrupts.

2. (Original) The processing system as recited in claim 1 wherein said plurality of first interrupts comprise:

hardware interrupts; and

software interrupts.

3. (Original) The processing system as recited in claim 2 wherein said hardware interrupts comprise:

a performance counter interrupt; and

a timer interrupt.

4. (Original) The processing system as recited in claim 1 wherein said core executes instructions.

5. (Original) The processing system as recited in claim 1 further comprising:

an interrupt controller, coupled to said plurality of second interrupts, for providing said plurality of second interrupts to said priority encoder with predefined interrupt priorities.

6. (Original) The processing system as recited in claim 1 wherein said priority encoder produces an indication of which of said first and second pluralities of interrupts has the highest priority.

7. (Original) The processing system as recited in claim 6 further comprising:

a vector generator, coupled to said priority encoder, for receiving said indication, and for producing an interrupt vector corresponding to said interrupt having the highest priority.

8. (Canceled)

9. (previously amended) The processing system as recited in claim 1 wherein said plurality of configurable priority registers are writable by the processing system.

10. (currently amended) A microprocessor for handling interrupts, the microprocessor receiving first interrupts from an interrupt controller, the first interrupts having architecturally fixed interrupt priorities, the microprocessor comprising:

a core, for executing instructions, said core generating second interrupts;

priority storage logic coupled to said core, for storing programmable priorities for said second interrupts~~, said programmable priorities being different than priorities for the first interrupts~~; and

a priority encoder, coupled to said core, and to said priority storage logic, for receiving the first and said second interrupts, and for prioritizing the first and said second interrupts utilizing ~~said~~the architecturally fixed interrupt priorities for the first interrupts, and said programmable priorities stored in said priority storage logic for said second interrupts.

11. (Original) The microprocessor as recited in claim 10 wherein said interrupt controller receives a plurality of third interrupts from sources thereof, prioritizes said third interrupts, and provides the prioritized third interrupts to the microprocessor as the first interrupts.

12. (Original) The microprocessor as recited in claim 11 wherein the first interrupts are presented to the microprocessor on first interrupt signal lines attached to the microprocessor.

13. (previously amended) The microprocessor as recited in claim 10 wherein said instructions executed by said core comprise:

first instructions for handling the first interrupts;

second instructions for handling said second interrupts; and

third instructions for storing said programmable priorities into said priority storage logic.

14. (Original) The microprocessor as recited in claim 10 wherein said second interrupts generated by said core comprise:

hardware interrupts; and

software interrupts.

15. (Original) The microprocessor as recited in claim 14 wherein said hardware interrupts comprise:

a performance counter interrupt; and

a timer interrupt.

16. (previously amended) The microprocessor as recited in claim 10 wherein said priority storage logic comprises:

a plurality of interrupt priority fields, each of said fields corresponding to one of said second interrupts.

17. (Original) The microprocessor as recited in claim 16 wherein each of said plurality of interrupt priority fields comprise:

a 4-bit field for storing one of sixteen distinct interrupt priorities.

18. (previously amended) The microprocessor as recited in claim 10 wherein said priority storage logic is located within a privileged resource within the microprocessor.

19. (Original) The microprocessor as recited in claim 18 wherein said privileged resource is programmable only when the microprocessor is executing privilege level, kernel mode, instructions.

20. (Original) The microprocessor as recited in claim 10 wherein the first interrupts have eight distinct priority levels.

21. (Original) The microprocessor as recited in claim 20 wherein said second interrupts have at least nine distinct priority levels that overlap the priority levels for the first interrupts.

22. (Original) The microprocessor as recited in claim 10 wherein said priority encoder, when prioritizing the first and said second interrupts, also uses priorities for the first interrupts established by the interrupt controller.

23. (Original) The microprocessor as recited in claim 10 wherein said priority encoder produces an indication of which of the first and said second interrupts has the highest priority.

24. (Original) The microprocessor as recited in claim 23 further comprising:

a vector generator, coupled to said priority encoder, for receiving said indication, and for producing an interrupt vector corresponding to a highest priority interrupt.

25. (previously amended) The microprocessor as recited in claim 24 further comprising:

programmable offset storage logic, coupled to said vector generator, for providing a programmed offset to said vector generator to allow said vector generator to produce said interrupt vector.

26. (previously amended) The microprocessor as recited in claim 25 wherein said programmable offset storage logic is programmed with said programmed offset by kernel mode instructions executing on the microprocessor.

27. (currently amended) A method for prioritizing on-core and off-core interrupts within a processing system, comprising:

receiving the off-core interrupts, the off-core interrupts having architecturally fixed interrupt priorities;

receiving the on-core interrupts, the on-core interrupts having programmable priority levels which are intermediate to the architecturally fixed interrupt prioritiespriority levels for the off-core interrupts;

sorting the received off-core and on-core interrupts according to their priority levels; and

producing an indication of which of the received off-core and on-core interrupts has the highest priority.

28. (Original) The method as recited in claim 27 wherein the on-core interrupts comprise:

a performance counter interrupt; and

a timer interrupt.

29. (Original) The method as recited in claim 27 wherein the off-core interrupts are initially prioritized by an interrupt controller.

30. (Original) The method as recited in claim 27 wherein said sorting comprises:

examining the priority levels of each of the received off-core interrupts;

examining the programmable priority levels of each of the received on-core interrupts; and

selecting one of the received on-core or off-core interrupts with the highest priority level.

31. (Original) The method as recited in claim 30 wherein said producing comprises:

receiving the one of the interrupts with the highest priority level;

examining a programmable offset; and

calculating an interrupt vector for the one of the interrupts with the highest priority level utilizing said programmable offset.

32. (Original) The method as recited in claim 31 further comprising:

causing the processing system to jump to the interrupt vector.

33. (Canceled)

34. (Canceled)

35. (Canceled)

36. (Canceled)

37. (Original) A computer data signal embodied in a computer readable transmission medium comprising:

computer-readable program code for providing a microprocessor for handling both internally generated and externally generated interrupts, the externally generated interrupts having architecturally fixed interrupt priorities, said program code comprising:

first program code for providing a programmable offset storage register for storing a programmable offset;

second program code for providing a plurality of interrupt priority level fields that store programmed priority levels for the internally generated interrupts;

third program code for providing a priority encoder, the encoder prioritizing between the internally generated and externally generated interrupts utilizing the programmed priority levels for the internally generated interrupts, and the fixed interrupt priorities for the externally generated interrupts, to select an interrupt with the highest priority, the encoder providing an indication of the interrupt with the highest priority; and

fourth program code for providing a vector generator for generating an interrupt vector for the interrupt with the highest priority, the vector generator utilizing the programmable offset to calculate the interrupt vector.

38. (Original) The computer data signal as recited in claim 37 wherein the programmable offset storage register is programmed with the programmable offset by kernel mode instructions.

39. (Original) The computer data signal as recited in claim 37 the externally generated interrupts have up to eight distinct priority levels and the internally generated interrupts have up to nine overlapping priority levels.